




Article

Machine-Learning-Based Darknet Traffic Detection System for IoT Applications

Qasem Abu Al-Haija ^{1,*} , Moez Krichen ^{2,3}  and Wejdan Abu Elhaija ⁴ 

¹ Department of Computer Science/Cybersecurity, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan

² Department of Computer Science, Al-Baha University, Al Baha 3029, Saudi Arabia; moez.krichen@redcad.org
³ ReDCAD Laboratory, National School of Engineers of Sfax, University of Sfax, Sfax 3038, Tunisia

⁴ Department of Electrical Engineering, Princess Sumaya University for Technology (PSUT), Amman 11941, Jordan; elhaija@psut.edu.jo

* Correspondence: q.abualhaija@psut.edu.jo

Abstract: The massive modern technical revolution in electronics, cognitive computing, and sensing has provided critical infrastructure for the development of today's Internet of Things (IoT) for a wide range of applications. However, because endpoint devices' computing, storage, and communication capabilities are limited, IoT infrastructures are exposed to a wide range of cyber-attacks. As such, Darknet or blackholes (sinkholes) attacks are significant, and recent attack vectors that are launched against several IoT communication services. Since Darknet address space evolved as a reserved internet address space that is not contemplated to be used by legitimate hosts globally, any communication traffic is speculated to be unsolicited and distinctively deemed a probe, backscatter, or misconfiguration. Thus, in this paper, we develop, investigate, and evaluate the performance of machine-learning-based Darknet traffic detection systems (DTDS) in IoT networks. Mainly, we make use of six supervised machine-learning techniques, including bagging decision tree ensembles (BAG-DT), AdaBoost decision tree ensembles (ADA-DT), RUSBoosted decision tree ensembles (RUS-DT), optimizable decision tree (O-DT), optimizable k-nearest neighbor (O-KNN), and optimizable discriminant (O-DSC). We evaluate the implemented DTDS models on a recent and comprehensive dataset, known as the CIC-Darknet-2020 dataset, composed of contemporary actual IoT communication traffic involving four different classes that combine VPN and Tor traffic in a single dataset covering a wide range of captured cyber-attacks and hidden services provided by the Darknet. Our empirical performance analysis demonstrates that bagging ensemble techniques (BAG-DT) offer better accuracy and lower error rates than other implemented supervised learning techniques, scoring a 99.50% of classification accuracy with a low inferencing overhead of 9.09 μ second. Finally, we also contrast our BAG-DT-DTDS with other existing DTDS models and demonstrate that our best results are improved by (1.9~27%) over the former state-of-the-art models.

Keywords: cybersecurity; machine learning; Internet of Things (IoT); IDS system; networks; darknet; blackhole; ensemble learning



check for updates

Citation: Abu Al-Haija, Q.; Krichen, M.; Abu Elhaija, W. Machine-Learning-Based Darknet Traffic Detection System for IoT Applications. *Electronics* **2022**, *11*, 556. <https://doi.org/10.3390/electronics11040556>

Academic Editors: Marcello Traiola, Elena-Ioana Vătăjelu and Angeliki Kritikakou

Received: 28 December 2021

Accepted: 10 February 2022

Published: 12 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

IoT and other communication technologies have dramatically improved our ability to comprehend our environment. Life quality may be improved through the use of IoT technologies, which have the potential to gather and analyze data about the surrounding environment [1]. This circumstance facilitates the development of smart cities by making it easier for things and humans to communicate with each other. There were an estimated 50 billion Internet of Things (IoT) devices by the end of 2020 [2,3].

The IoT is a sophisticated and interconnected system. As a result, it is difficult to meet the security requirements of an IoT system with a large attack surface. The widespread use

of IoT has had the unintended consequence of making IoT deployment an interconnected process. There are a number of considerations to keep in mind while deploying an IoT system: security, energy efficiency, analytics approaches, and interoperability with other software applications [3]. The IoT devices, on the other hand, often operate in the absence of a human operator. These devices can therefore be physically accessed by an intruder. Intruders can gain access to private information via eavesdropping on wireless networks used by IoT devices, which are often connected by a communication channel.

As the IoT area continues to evolve, defining a reference architecture that can accommodate both present functionality and future enhancements will be a significant task. As a result, such an architecture must be: scalable, in order to handle a rising number of devices and services without compromising their performance; interoperable, so that devices from different manufacturers may collaborate to accomplish shared objectives; distributive, in order to enable the development of a distributed environment in which data are processed by different entities in a distributed manner after being acquired from various sources; and capable of operating with minimum resources [4].

There is currently no single reference architecture, and building one is proving difficult despite several standardization initiatives. The fundamental issue is the inevitable fragmentation of possible applications, each of which is dependent on a plethora of frequently disparate factors and design standards. This issue must be combined with each supplier's desire to promote its platform for comparable applications [4,5]. Figure 1 depicts some of the most often seen Internet of Things architectures [5].

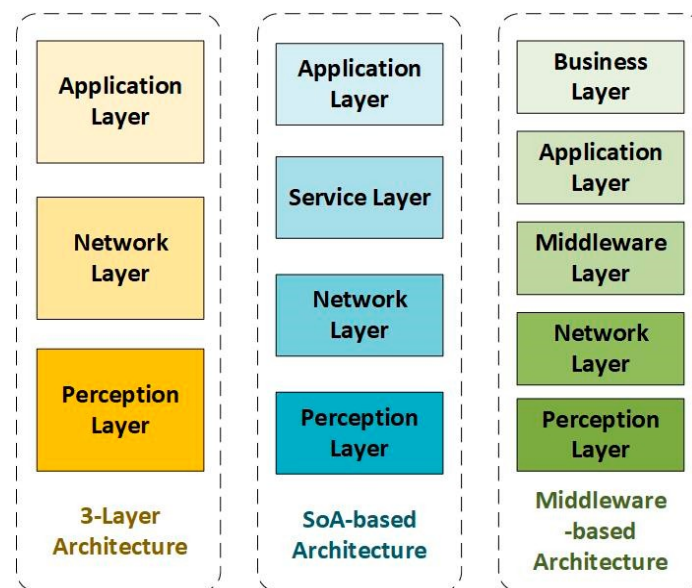


Figure 1. Most popular IoT architectures.

Cyber-attacks against IoT devices are increasing in line with their growth, and IoT devices can be tricked into becoming bots that mindlessly obey orders in order to commit crimes as part of a botnet. A botnet is a network of infected devices used by an attacker to carry out tasks such as DDoS attacks, Bitcoin mining [6], and spam email distribution. Almost any internet-connected device may become infected and join a botnet. Because IoT devices have poor security and are thus simple candidates for infection, they are frequently enlisted to become bots. Botnets, like other malware, may be found on Darknet marketplaces. Botnets may be hired, and botnet source codes can be purchased or even leaked, as with the Mirai botnet. Prices range from tens to hundreds of dollars based on the type of service, the number of bots/devices accessible for usage in the botnet, and the power and time of the DDOS attack [7–9]. Some botnets compete against one another due to the competitive nature of the Darknet. If an IoT device has previously been attacked, another botnet can try to replace the infection with its own program and, in certain situations,

“fix” the security issue utilized by the prior botnet to prevent re-infection and maintain its position on the susceptible device.

The Dark Web, Deep Web or Darknet describes a network of websites that are open to the public but conceal the servers’ IP addresses. Deep Web is expected to be many orders of magnitude larger than surface web [10]. This number has only risen since then, due to the Internet’s rapid expansion. Navigating the Darknet layer necessitates the use of specialized tools and applications. It cannot be accessed without them. The Darknet, on the other hand, is where consumers are likely to find their stolen files and compromised data for sale, as well as any other unauthorized product for sale. This layer is built on peer-to-peer computer networking and routing communications via so much uncertainty that monitoring is almost impossible. That is why there are so many unauthorized activities taking place within it. One of the configurations on which the Darknet is built is a Tor network [11], which employs an onion routing protocol. Tor software allows users to connect to the Darknet and communicate/browse over huge peer-to-peer connections. To offer security, large obscurity and constantly changing channels, circuits, and connections are employed. Figure 2 depicts the basic concept of Darknet utilization of TorNet. According to the figure, in the onion routing protocol, the data to be sent are encapsulated in encryption layers, similar to the layers of an onion [11]. The encrypted data are then sent via a succession of network nodes known as onion routers, each of which “peels” away (or decrypts) a single layer of encryption, revealing the data’s next destination. The data are delivered to their destination when the last layer is decrypted. Because each intermediate only knows the position of the nodes immediately before and after it, the sender remains anonymous [11].

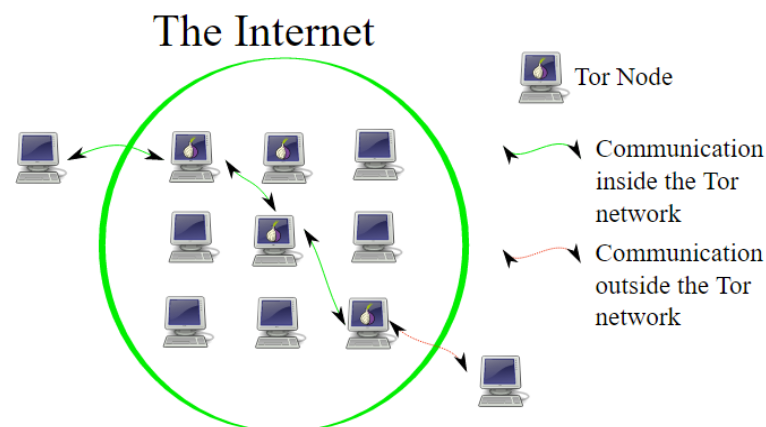


Figure 2. Most popular IoT architecture.

Because of the Darknet’s huge scale and reach, more effective measures for mitigating its potential threats are required. Modern methods must be utilized to trace down the black market and its transactions in order to grasp the criminals. The unindexed, fragmented, and multi-layered structure of the Darknet makes it more difficult to locate the criminals therein. Forensic law enforcement agencies want strong digital evidence to overcome the problems of finding and prosecuting criminals in the Dark Web environment. There are several relevant studies on IoT in the literature that address various areas of machine learning in cybersecurity. Many of the latest cybersecurity solutions were created by combining machine learning with cybersecurity. The majority of new IoT intrusions are minor modifications from previously known vulnerabilities [12]. These little variations in IoT attacks have been shown to be difficult to identify/classify using typical machine-learning approaches. Deep neural networks have been used in promising state-of-the-art cybersecurity research [13–17]. Table 1 highlights the standard and traditional machine-learning research [18].

Table 1. Examples of several machine-learning approaches.

Learning Type	Model Building	Examples
Supervised	Algorithms or models learn from labeled data (Task-Driven Approach)	Classification, regression
Unsupervised	Algorithms or models learn from unlabeled data (Data-Driven Approach)	Clustering, associations, dimensionality reduction
Semi-supervised	Models built using combined data (Labeled + Unlabeled)	Classification, clustering
Reinforcement	Models based on reward or penalty (Environment-Driven Approach)	Classification, control

The improved IoT security (based on IoT-specific threats) methodology used in a few studies [19–23] gave a comprehensive background on all IoT areas and provided a comprehensive background on all IoT areas. Those studies did not, however, explore the areas of Darknet-based IDS for IoT in the same way that we did. The study in Table 2 demonstrates the important nature of anomaly-based intrusion detection systems for the security of things [24]. To help illustrate their solution context and process, this table shows gaps in prior survey studies inside the conventional architectural layers of IoT systems and links them with IDSs, such as anomaly-based IDS.

Table 2. Detection methodology characteristics for IoT IDS.

Detection Methods	Disadvantages
Statistics-based: examines network traffic and processes the data using complex statistical techniques.	<ul style="list-style-type: none"> • Requires a high level of statistical expertise • Is simple but less precise • Is real time
Pattern-based: identifies the characters, forms, and patterns in the data.	<ul style="list-style-type: none"> • Easy to implement • A hash function could be used for identification.
Rule-based: use an attack “signature” to identify unusual network activity.	<ul style="list-style-type: none"> • Because rules need pattern matching, the computational cost of rule-based systems may be rather expensive. • It is extremely difficult to predict which activities will occur and when. • It needs a huge number of rules in order to counteract all conceivable attacks. • A low number of false positives • A high detection rate
State-based: examines a sequence of events in order to ascertain the possibility of an attack.	<ul style="list-style-type: none"> • Probabilistic, self-training • Low probability of false positives
Heuristic-based: identifies any abnormal activity that is not consistent with the norm.	<ul style="list-style-type: none"> • It needs knowledge and experience. • Experimental and evolutionary learning

While improved intelligent intrusion detection systems for IoT networks have been significantly investigated and proposed in the literature, only a few studies have explored the areas of Darknet intrusion (blackholes or sinkholes) detection systems for IoT networks. Therefore, in this paper, we propose an efficient machine-learning-based system for identifying and classifying Darknet traffic for IoT networks. Specifically, the present article introduces an ensemble learning-based model for detecting and characterizing VPN and Tor applications as the true representatives of Darknet traffic by combining two public datasets from the Canadian Institute for Cybersecurity (CIC), namely ISCXTor2016 and ISCXVPN2016, to create a complete Darknet dataset covering Tor and VPN traffic, namely CIC-Darknet2020 [25]. Six different machine-learning techniques were implemented and evaluated using CIC-Darknet2020, including bagging decision tree ensembles (BAG-DT), AdaBoost decision tree ensembles (ADA-DT), RUSBoosted decision tree ensembles (RUS-DT), optimizable decision tree (O-DT), optimizable k-nearest neighbor (O-KNN), and optimizable discriminant (O-DSC). Our empirical results show that the BAG-DT pre-

diction model outperforms the other mentioned models and all existing state-of-the-art machine-learning-based models for Darknet activities/instructions detection and classification. Specifically, our best performance results are registered in a DTDS-based BAG-DT, scoring a maximum detection/classification accuracy of 99.50% with a low inferencing overhead of 9.09 μ Second.

1.1. Summary of Our Contributions

Our main contributions can be summarized as follows:

1. We developed a multi-purpose and high-performance anomaly-based IoT DIDS utilizing several supervised machine-learning approaches.
2. We differentiate and measure the performance of six supervised learning methods (BAG-DT), (ADA-DT), (RUS-DT), (O-DT), (O-KNN), and (O-DSC) for IoT DIDSs using the CIC-Darknet2020 datasets.
3. We present a comprehensive experimental evaluation of six different ML techniques using ten typical systems of measurement factors.
4. We contrast our findings with state-of-the-art approaches and show that our BAG-DT-based DIDS is better than existing studies by 1.9–20% in the same area of study.

1.2. Paper Organization

The rest of this paper is organized as follows: Section 2 surveys several up-to-date-related studies for machine-learning-based Darknet traffic detection systems (DTDS) in IoT networks. Section 3 describes the system, design, development, and evaluation phases, as well as development environments and configurations. Section 4 presents and discusses our experimental results of the proposed ML-DTDS-IoT. Finally, Section 5 concludes and summarizes the paper's findings and provides recommendations for future research directions.

2. Related Work

The internet security sector now places a strong emphasis on cyberspace surveillance in order to generate cyber intelligence. The authors of [26] presented an interesting survey on the Darknet in which they gave a taxonomy in connection to Darknet technologies, presented evaluations and examination of Darknet information, compared Darknet to other trap-based monitoring systems, and highlighted open research issues related to the Darknet.

2.1. First Works for Traffic Classification

Studies analyzing Darknet traffic for diverse objectives abound in the literature. Early efforts from the early 2000s used decision trees [27], support vector machines [28], Bayesian analysis techniques [29], profile hidden Markov models [30], and other clustering algorithms [31]. More precisely, the authors of [27] discussed a method for classifying server traffic using decision trees established during the training step. The trees were extracted from the traffic, which was described using a number of features developed by the authors to represent stream behavior. In [28], an efficient in-the-dark traffic classification of typical application protocols for TCP sessions was proven using support vector machines (SVMs). In this context, aggregate features were used to characterize each flow. The authors of [29] demonstrated the accuracy and trust of the traffic classification that resulted from the application of Bayesian Analysis Techniques. They demonstrated that, in its most generic definition, a Naive Bayes classifier can achieve 65% accuracy for data from the same interval of time and that, when paired with a number of simple modifications, it can attain over 95% accuracy. They also demonstrated that their methodology is temporally stable by comparing test and training sets distanced by more than a year. The authors of [30] described a strategy based on profile hidden Markov models that involved creating statistical models for the sequence of packets created by every protocol of interest and then using these models to determine the protocol being used. The authors of [31] examine twelve different clustering methods. The review discusses the contributions of these twelve

techniques to clustering methodologies, as well as current issues and recommendations for future traffic flow clustering research.

2.2. VPN Traffic Classification

Virtual private networks (VPNs) [32] are a type of technology that allows communicating securely over an unsafe network. VPN solutions are divided into multiple varieties, each with its own approach to security, benefits, drawbacks, and reliance on various protocols and standards. IPsec, PPTP, and TLS are the three main varieties. According to [33], the purpose of VPN traffic classification is to overcome the constraints and problems posed by their structure and the hostile environment in which they operate. This will enable the development of a better routing solution, one that avoids the flaws in standard routing protocols that control VPN operations. The authors of the work presented in [33] proposed a MATLAB implementation that uses artificial neural networks and time-related features on Apache Spark [34] to classify VPN network data flow. The results obtained in this work were as follows: the VPN identification accuracy was about 96.7%, and the non-VPN accuracy was about 92.5%. In [35], the dataset ISCXVPN2016 [36], as well as six machine learning models, were used to find the best supervised model for distinguishing VPN from non-VPN traffic. All other supervised models were shown to be inferior to gradient boosting tree (GBT) and random forest (RF). The authors of [36] used the same database to train multi-class classifiers that accurately classified VPN traffic into seven different categories using k-nearest neighbor (kNN) and the c4.5 decision tree technique. The classification accuracy obtained was around 80%. In order to develop a model capable of discriminating between VPN and non-VPN traffic in real time, the authors of [37] used machine-learning-based techniques on a multi-layer perceptron neural network model. To create a representative dataset of VPN and non-VPN values, real network data were collected using a variety of technologies. The packet capture for this dataset was performed with Wireshark [38]. The observed results suggest that the neural network's overall detection accuracy in the post-training test was around 94%.

2.3. Tor Traffic Classification

Tor [39] is the most widely used privacy-enhancing tool at the moment. By encrypting and tunneling communications over a distributed network of servers known as Tor nodes, it can hide users' identities and internet activities. According to [40], Tor traffic classification seeks to improve performance by identifying multiple service classes for its traffic. Indeed, even though interactive web browsing accounts for the vast majority of Tor traffic, bulk downloading consumes an inordinate amount of Tor's limited capacity. DiffTor [40] is a real-time mechanism for classifying Tor's encrypted circuits by application and assignment of separate service classes to each. The results of the experiments showed that they were able to classify circuits with more than 95% accuracy and that the proposed classification, mixed with QoS, resulted in a 75% gain in responsiveness and an 86% reduction in download duration for interactive users. The authors of [41] presented a multi-level Tor traffic classification and identification framework based on network flow features [42] for both mobile and PC platforms. For the mobile platform, they found that time-related features have a greater impact than non-time-related features, whereas, for the PC platform, the converse is true. To distinguish Tor anonymous traffic from mixed traffic, a hierarchical classification strategy based on an enhanced decision tree algorithm was suggested in [43], and then the TriTraining algorithm [44] was used to partition the identified anonymous traffic. The experiments demonstrate that Tor anonymous traffic is recognized at a rate of more than 99%, with classification accuracy reaching 94%. The authors of [45] conducted a thorough analysis of Tor traffic classification, quantification, and comparison of various strategies for deanonymization, path selection, and increasing the performance of encrypted communication in the Darknet.

2.4. Use of Neural Networks in Recent Works

Advances in artificial intelligence have enabled digital systems to detect and identify Darknet activity on their own. A generalized strategy for detecting and categorizing Darknet traffic using Deep Learning was proposed in [46]. To recognize network traffic more correctly, the researchers used adapted convolution long short-term memory and extreme gradient boosting as feature selection techniques. The results show that the suggested approaches detect and categorize Darknet traffic with an accuracy of 89% for categorization and 96% for detection. To identify Darknet traffic, ref. [47] used popular machine-learning classification techniques. A receiver operating characteristics (ROC) analysis was combined with a feature significance analysis for the best classifier. The studies used the new dataset CICDarknet2020, and the classifiers were trained to classify binary and multi-class data. The random forest method was used to produce an average prediction accuracy of over 98%. Using ensemble machine-learning algorithms on the CIC-Darknet2020 dataset, ref. [48] was able to differentiate Darknet traffic from benign traffic with 98% accuracy. Furthermore, with 97% accuracy, the researchers recognized the sort of program running beneath the Darknet traffic. They also used a game-theoretic method to demonstrate the impact of selected features and interpret the output of machine-learning models in order to better understand Darknet traffic behavior. The study presented in [49] proposes a weight-agnostic neural network framework for Darknet traffic and network management, with the goal of automating the suspicious intent recognition process in real time. According to the authors, the presented approach allows for a more customized pattern recognition system to cope with changing situations without previous training. The study proposed in [50] presents a CNN-based classification system that can recognize both protocols and applications. The proposed method employs a two-stage, two-label classification system. The protocol used for encrypted traffic is classified in the first step. The second stage employs the corresponding classifier to categorize applications based on the traffic protocol. On the CICDarknet2020 dataset, experimental findings reveal that the suggested approach yields an accuracy of about 97.6%.

2.5. Summary of Surveyed Research Works

To improve readability and to provide more insights into the surveyed paper in this research, we end this section by summarizing some of the most recent and relevant studies reported in the literature, as shown in Table 3.

Table 3. A summary of the investigated and surveyed research works.

Ref	Year	Technique	Contribution
[27]	2003	Decision Trees	Behavioral authentication of server flows and classification of server traffic
[28]	2008	Support Vector Machines	Efficient in-the-dark traffic classification of typical application protocols for TCP sessions
[29]	2005	Bayesian Analysis Techniques	Increasing the accuracy of the Bayes Classifier through a set of simple modifications
[30]	2006	Profile Hidden Markov Models	Creating statistical models for the sequence of packets created by every protocol of interest and using these models to determine the protocol being used
[31]	2006	Clustering Algorithms	Review of 12 clustering methodologies, current issues, and recommendations for traffic flow clustering research
[33]	2020	Artificial Neural Networks and Time-Related Features	Classifying VPN network data flow using ANNs and Time-Related Features
[35]	2017	Six Machine Learning Techniques	Distinguishing VPN from non-VPN traffic and proving that Gradient Boosting Tree and Random Forest are the best machine-learning techniques to use
[36]	2016	K-Nearest Neighbor and C4.5 Decision Tree	Creating multi-class classifiers that accurately classify VPN traffic into seven different categories

Table 3. Cont.

Ref	Year	Technique	Contribution
[37]	2018	Multi-Layer Perceptron Neural Network and Wireshark	Building a representative dataset of VPN and non-VPN values and classifying VPN network data
[40]	2012	Classification Techniques Mixed With QoS	Real-time classification of Tor's encrypted circuits by application and assignment of separate service classes to each
[41]	2020	Network Flow Features	Multi-level Tor traffic classification and identification framework for both mobile and PC platforms
[43]	2017	Decision Tree and TriTraining Algorithm	A hierarchical classification strategy for distinguishing Tor anonymous traffic from mixed traffic
[45]	2018	Various Techniques	A thorough analysis of Tor traffic classification, quantification, and comparison of various strategies
[46]	2021	Convolution-LSTM and Extreme Gradient Boosting	A generalized strategy for detecting and categorizing Darknet traffic using Deep Learning
[47]	2021	ML and Receiver Operating Characteristics	A feature significance analysis for the best classifier of binary and multi-class data
[48]	2021	ML and Game-Theoretic Method	Differentiating Darknet traffic from benign traffic using ensemble machine-learning algorithms
[49]	2021	Weight-Agnostic Neural Network	Framework for Darknet traffic management for automating the suspicious intent recognition in real time
[50]	2021	Convolutional Neural Network	Two-stage, two-label classification system that can recognize both protocols and applications

3. System Modeling and Environment

This research aims to develop a Darknet IDS system that can detect Darknet activities of common IoT cyber-attacks using supervised machine-learning methods. Thus, the system of interest (SOI) in this research is concerned with developing an empirical system at the IoT application layer to detect these Darknet cyber-attacks. Specifically, once the representative data are accumulated, the SOI is composed of three units illustrated in Figure 3: feature engineering unit to handle preprocessing and encoding for the collected Darknet traffic dataset, learning models unit to train and test the various implemented machine-learning algorithms using the target dataset and pick up the best-optimized model, and traffic classification unit to evaluate system performance via several metrics, such as accuracy, precision, false alarms, and others. These units are used to produce categorization for every traffic record of the Darknet dataset through the four-class classifier.

3.1. The Darknet Traffic Dataset

Darknet, also known as blackhole or sinkhole attacks, are significant, and recent attack vectors were launched against several IoT communication services [51]. Since Darknet address space evolved as a reserved internet address space that is not contemplated to be used by legitimate hosts globally, any communication traffic is speculated to be unsolicited and distinctively deemed a probe, backscatter, or misconfiguration. The main objective of this research is to detect Darknet traffic to combat suspected activities before they assault the cyber world. To address the problem stated in this research, one should first consider collecting a representative traffic dataset that the proposed model can utilize to express the Darknet traffic over the IoT network communications. A CIC-DarkNet-2020 dataset, which was compiled by Arash et al. (2020) [25] and made public by the Canadian center of cybersecurity (CIC), has been utilized in this research. The CIC-DarkNet-2020 dataset is a novel and inclusive dataset that intelligently combines the traffic records of two publicly available datasets, (ISCXVPN2016 and ISCTXor2017 [52]), to produce a comprehensive dataset for Darknet traffic activities covering a wide range of Darknet activities, including VPN and Tor traffic.

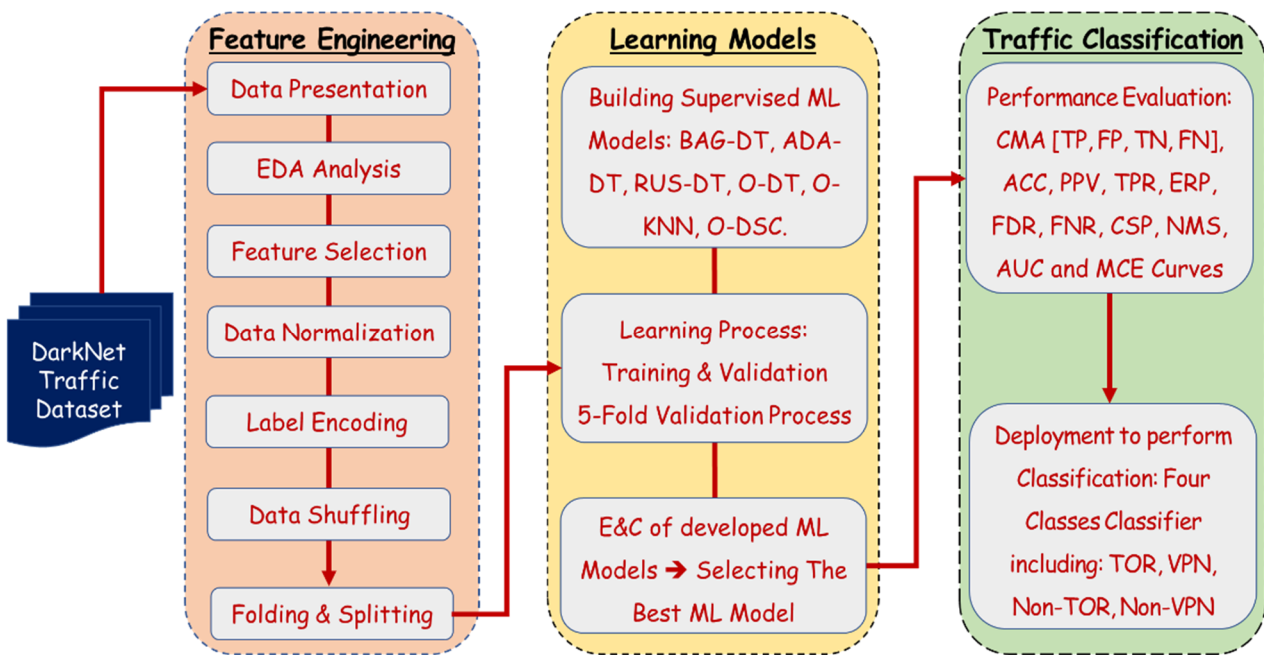


Figure 3. System development model diagram for proposed Darknet-IDS of IoT (DTDS-IoT) network traffic via ML techniques.

CIC-DarkNet-2020 dataset recorded a total of ~141,530 network traffic records comprising benign traffic (~11,7219) and Darknet traffic (~24,311). Darknet traffic activities are meant to target the desecration of several significant IoT and networking services, including audio/video streaming services (such as YouTube), browsing services (such as Firefox), chatting and VOIP services (such as Skype), email services (such as SMTP), peer-to-peer (P2P) services (such as BitTorrent), and file transfer services (such as FTP). A summary of CIC-DarkNet-2020 dataset distribution using four classes (VPN, TOR, Non-VPN, or Non-TOR), is provided in Figure 4 below [25].

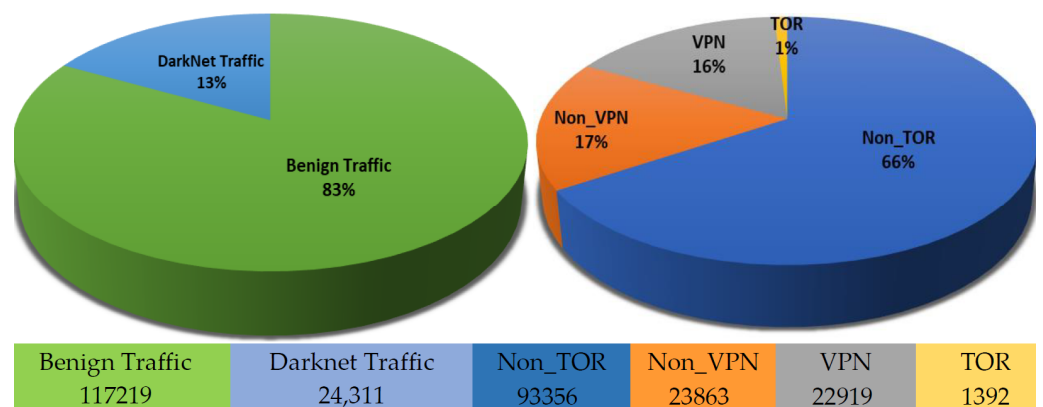


Figure 4. CIC-DarkNet-2020 dataset sample distribution.

3.2. Feature Engineering Unit

Feature engineering is the selection, manipulation, and transformation of raw data into attributes that may be fed into the machine-learning algorithms for further processing, training, validation, and prediction using a pipeline of preprocessing stages started at the data accumulation process. Our feature engineering unit performs the following consecutive preprocessing operations before the data reach the learning models:

- **Data Presentation:** CIC-DarkNet-2020 dataset is initially available in CSV format. Therefore, to be processed by the MATLAB platform, at the outset, it should be

imported from the CSV file and presented as a table of data records in the MATLAB tables with named columns and numbered rows.

- Exploratory data analysis (EDA): EDA of the dataset performs vital data curation tasks to gain a deeper insight into the dataset. Such a process completes a preliminary enhancement process of the dataset by checking missing data values and providing proper substitution for the missed records, replacing null values with appropriate replacements, such as zeroes, visualizing the dataset classes' histogram to gain more insights into the classes and features.
- Feature Selection: Datasets are comprised of several features with diverse datatypes. Nevertheless, not all features can be considered for machine-learning models, since they can either be unlearnable (such as string features) or might have a negative impact on the classifier performance. The coefficient score approach is employed to extract the most influential features of the CIC-DarkNet-2020 dataset to obtain the best features that can be used later in training and validating the learning models.
- Data Normalization: normalization is usually performed over the scattered data points with a significant range between the points. Therefore, normalization is performed in order to re-scale data points to be in the same range and significance (usually 0–1). This will disallow the larger values from dominating other data points in the dataset. Therefore, we apply min–max normalization at the stage of preprocessing to have all numerical data within a range between 0 and 1. The min–max normalization of a datapoint D_i within a set of points (D), is given by the following formula (D_i^{Norm}):

$$D_i^{Norm} = \frac{[D_i - \min(D)]}{[\max(D) - \min(D)]} \quad (1)$$

- Label Encoding: Label encoding techniques are utilized to convert categorical data into numerical data that may be processed by machine-learning methods.
- This research employed integer encoding techniques to represent the categorical data as a numerical record. For instance, the output class labels were encoded as {non-Tor: 00, non-VPN: 01, Tor: 02, and VPN: 03}.
- Data Shuffling: The shuffling process is a preprocessing operation conducted over the dataset samples (rows) by randomly rearranging data from a dataset to produce a new arrangement for the dataset that can be safely used for ML testing and training, without having the classifier being biased to any of the underlying classes. This will guarantee anonymity while ensuring data statistics are kept exactly the same. Figure 5 illustrates the data shuffling process.
- Folding and Splitting: To ensure a high level of the validation process of the proposed predictive models, we have conducted a k-fold cross-validation operation incorporating five different folds (distributions) with data split into 70% for training and 30% for validation (testing). For every fold, a new validation experiment involves further data distribution to ensure that all data items have participated in the training and validation process. Our folding and splitting process is shown in Figure 6 and depicts the dataset distribution throughout the folds for each experiment.

3.3. Learning Models Unit

The learning process is a practice of applying algorithmic models to data in an iterative manner to enable the machine (such as a computer) to discover hidden patterns that it can use to make predictions. A large number of supervised machine-learning algorithms are usually employed to build models to address three common tasks: regression, prediction, and classification [53–55]. As our DTDS problem is modeled as a classification problem, we have used and implemented six different supervised machine-learning methods, including bagging decision tree ensembles (BAG-DT), AdaBoost decision tree ensembles (ADA-DT), RUSBoosted decision tree ensembles (RUS-DT), optimizable decision tree (o-dt), optimizable k-nearest neighbor (O-KNN), and optimizable discriminant (O-DSC). A summary of specifications and configurations for the implemented machine-learning models is pro-

vided in Table 4. After that, the developed model undergoes an evaluate-and-compare (E and C) process to pick up the best (optimum) ML technique to model the problem of DTDS. To do so, the classification accuracy metric is evaluated for each model and used as a vital metric to compare the developed models and select the best optimized model to be employed to address the Darknet traffic detection and classification task. The performance accuracy is used as a unified performance metric by which we might be able to provide some insights about developed methods and help select the best ML-based solution for this specific problem.

Table 4. Learning model specifications and configurations.

ML Model	Specifications
BAG-DT	Preset: Bagged Trees, Ensemble method: Bag, Learner type: Decision tree, Maximum number of splits: 89161, Number of learners: 30, Data Distribution Policy: 70% training and 30% testing, 5-Fold Cross-Validation.
ADA-DT	Preset: Boosted Trees, Ensemble method: AdaBoost, Learner type: Decision tree, Maximum number of splits: 20, Number of learners: 30, Learning rate: 0.1, Data Distribution Policy: 70% training and 30% testing, 5-Fold Cross-Validation.
RUS-DT	Preset: RUSBoosted Trees, Ensemble method: RUSBoost, Learner type: Decision tree, Maximum number of splits: 20, Number of learners: 30, Learning rate: 0.1, Data Distribution Policy: 70% training and 30% testing, 5-Fold Cross-Validation.
O-DT	Preset: Fine Tree, Maximum number of splits: 100, Split criterion: Gini’s diversity index, Surrogate decision splits: On, using a maximum of 10 surrogates, Data Distribution Policy: 70% training and 30% testing, 5-Fold Cross-Validation.
O-KNN	Preset: Optimizable KNN, Number of neighbors: 2, Distance metric: Euclidean, Distance weight: Squared inverse, Standardize data: false, Optimizer: Bayesian optimization, Acquisition function: Expected improvement per second plus, Iterations: 30.
O-DSC	Preset: Optimizable Discriminant, Discriminant type: Linear, Quadratic, Diagonal Linear, Diagonal Quadratic, Optimizer: Bayesian optimization Acquisition function: Expected improvement per second plus, Iterations: 30, 5-Fold Cross-Validation.

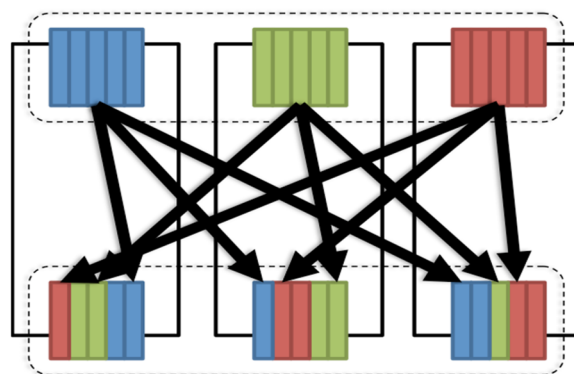


Figure 5. Data shuffling process illustration.

Experiment # 1	Training		Testing (30%)
Experiment # 2	Training		Testing (30%)
Experiment # 3	Training		Testing (30%)
Experiment # 4	Training	Testing (30%)	Training
Experiment # 5	Testing (30%)	Training	

Figure 6. Scheme for the folding and splitting of the dataset used in the proposed system.

3.4. Traffic Classification Unit

Once the optimum model is selected (based on performance accuracy), a further evaluation system of measurement is assessed for the best ML-DTDS-IoT. This includes investigating the confusion matrix that reports the number of true positive predicted samples, the number of true negative predicted samples, the number of false-positive predicted samples, and the number of false-negative predicted samples. Based on these parameters, several other performance evaluation measures can be computed, including classification accuracy rate (ACC%), positive predictive value (PPV%), true positive rate (TPR%), harmonic mean score (HMS%), classification error percent (ERP%), false discovery rate (FDR%), false negative rate (FNR%), and the number of misclassified samples (NMS#). Figure 7 summarizes the confusion matrix analysis with other performance evaluation measures mentioned. In addition, we report on the classification speed (CSF in samples per second) and area under the curve (AUC%).

		Predicted Classes		
		Positive	Negative	
Actual Classes	Positive	True Positive (TP)	False Negative (FN) Type II Error	Classification Accuracy (ACC) $ACC = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$ $ERP = (1 - ACC)\%$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Positive Predictive Value (PPV) $PPV = \frac{TP}{TP + FP} \times 100\%$ $FDR = (1 - PPV)\%$
		Harmonic Mean Score (HMS) $HMS = 2 \times \frac{PPV \times TPR}{PPV + TPR} \times 100\%$ $NMS = FP + FN$	True Positive Rate (TPR) $TPR = \frac{TP}{TP + FN} \times 100\%$ $FNR = (1 - TPR)\%$	

Figure 7. Confusion matrix with other performance evaluation measures.

After that, the selected model is extensively evaluated by a system of measurement mentioned earlier to gain more insights about the system model and the solution approach. The system is deployed to work as an actual application after ensuring the utilization of the optimum ML-based DTDS model that scores the optimum performance quality measures, particularly the classification accuracy, which is the vital evaluation metric. The system is utilized to provide a classification for the Darknet traffic into four output classes, including Non-Tor, Non-VPN, Tor, and VPN

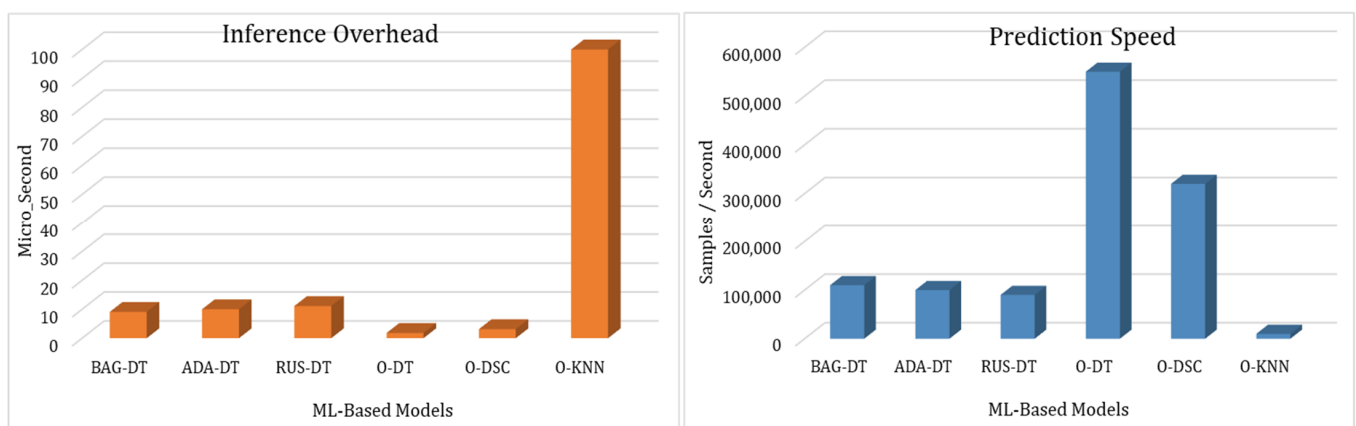
4. Results and Discussion

Based on the information mentioned above and the system architecture diagram, the proposed system has been developed, implemented, and evaluated using a high-performance computing platform (employing the 11th generation Intel Core i7 for fast processing operations and NVIDIA GeForce 4-GByte GPU for parallel computing operations) and built using MATLAB 2021b along with its accompanying learning and parallel computing tools. This section reports on the achieved investigational results of the DTDS model development using several machine-learning approaches, including several performance indicators. Table 5 contrasts the empirical results of classification accuracy and error rates obtained from modeling the DTDS-IoT using those, as mentioned earlier, six supervised machine-learning systems (BAG-DT, ADA-DT, RUS-DT, O-DT, O-DSC, O-KNN). This stage of experimentation and evaluation (i.e., results of Table 5) is a vital stage to characterize the performance of six machine-learning algorithms on this particular problem (i.e., machine-learning-based Darknet traffic detection systems (DTDS) in IoT networks) and select the optimum model for this specific problem.

Table 5. Experimental results were obtained from other machine-learning techniques.

	BAG-DT	ADA-DT	RUS-DT	O-DT	O-DSC	O-KNN
Accuracy %	99.5	95.4	93.9	97.3	83.6	97.1
Error %	0.5	4.6	6.1	2.7	16.7	3.9

In addition, Figure 8 contrasts the computational complexity of the employed machine-learning techniques in terms of prediction speed (measured in the number of samples per second) and the inference overhead (measured in microseconds). According to the figure, the most rapidly predictive models are the optimizable decision trees (O-DT) and the bagging decision trees (BAG-DT), scoring a prediction speed of 2.2×10^5 and 1.1×10^5 samples per second and the lowest inferencing overhead of $4.55 \mu\text{s}$ and $9.09 \mu\text{s}$, respectively. Conversely, the slowest predictive models are the optimizable k-nearest neighbor (O-kNN) and the optimizable discriminant (O-DSC), scoring a prediction speed of 0.7×10^5 and 0.1×10^5 samples per second and the lowest inferencing overhead of $14.28 \mu\text{s}$ and $100 \mu\text{s}$, respectively.

**Figure 8.** Computational complexity analysis for employed machine-learning techniques.

According to the classification accuracy proportions stated in Table 5 and the computational complexity in Figure 8, the DTDS-based bagging decision tree ensembles (BAG-DT) classifier outperforms other ML classifiers employed to develop the DTDS. Consequently, we emphasize our forthcoming analysis and discussion of the additional results obtained for the DTDS model via BAG-DT. We are also inclined to mention that BAG-DT could perform the two-class classification (normal vs. Darknet) with 100% accuracy. Since DTDS-BAG-DT is selected as the optimum model, we have traced its iterative learning process trajectory using the minimum classification error.

Figure 9 shows the performance analysis curve for the BAG-DT-based DTDS system. The minimum classification error as a cost function has been inspected for the BAG-DT classifier to follow the classifier state throughout 30 iterations of the learning process. This was conducted by tracing the minimum classification error (observed vs. estimated), targeting the best point for model hyperparameters. Subsequently, according to the figure, it can be seen that the best validation performance for the DTDS-BAG-DT system is achieved after iteration number 13 with minimum classification error values less than 0.005. The classification process remained saturated and stable for the rest of the iterations, recording no abnormality in the error analysis curves. Hence, the developed DTDS-BAG-DT system achieved a near-perfect performance, scoring an error rate approaching 0 (i.e., zero error performance stands for perfect models, while near-perfect models record error rates ≤ 0.01).

To gain more insight into the system development of the BAG-DT-based DTDS system and the solution approach, we have also investigated the overall four-class confusion matrix analysis for the DTDS-BAG-DT system, correlating the true classes vs. predicted classes, in Figure 10a. Based on the numbers reported by the matrix, it can be inferred

that the majority of CIC-Darknet-2020 traffic data were truly classified, i.e., the green cells of the confusion matrix (the number of truly classified samples = TP + TN = 140,862 out of 141,530 → 99.53%). In comparison, only a few (minority) of CIC-Darknet-2020 traffic data were falsely classified (the number of truly classified samples = FP + FN = 668 out of 141,530 → 0.47%). Additionally, Figure 10b depicts the four-class PPV-FDR matrix analysis of the DTDS-BAG-DT system for each class. The observation of this matrix demonstrates that all classes are precisely predicted with more than 99% precision (PPV) for each class, scoring an overall precision (PPV) for the DTDS-BAG-DT system of 99.45%. Moreover, Figure 10c depicts the four-class TPR-TNR matrix analysis of the DTDS-BAG-DT system for each class. The observation of this matrix demonstrates that three classes out of four (i.e., Non-Tor, Non-VPN, and VPN classes) have very high sensitivity (TPR) rates, recording more than 98.9% sensitivity (TPR) rates. Only one class out of four (i.e., Tor class) has a lower sensitivity rate of 89.9, rendering the overall sensitivity (TPR) for the DTDS-BAG-DT system of 96.93%. Thus, the DTDS-BAG-DT system’s overall performance can be described as accurate, precise, and sensitive in providing both Darknet detection and classification for the IoT network traffic network.

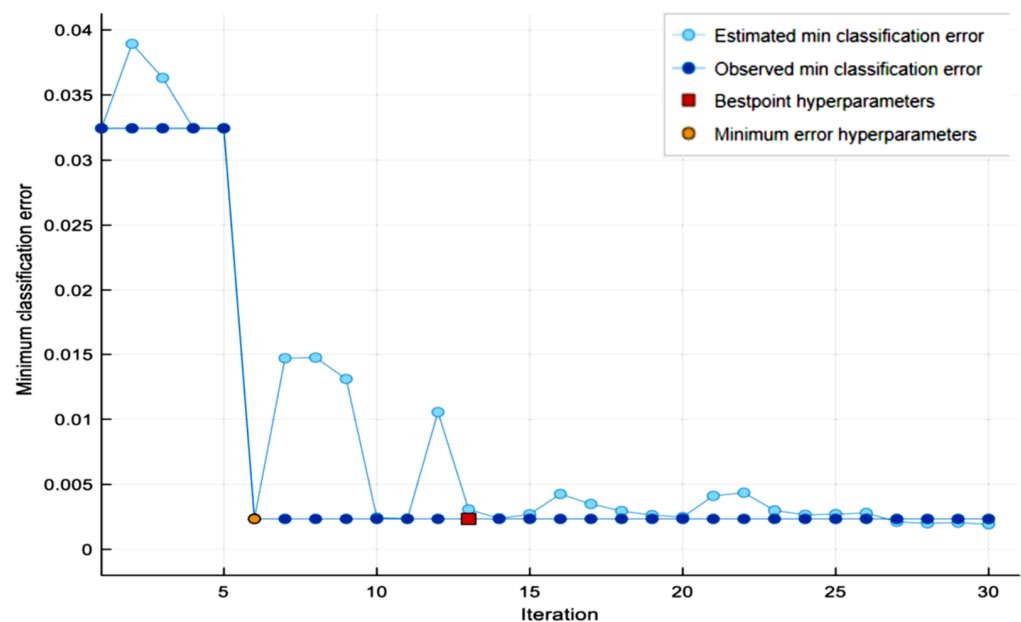


Figure 9. Learning process trajectory for BAG-DT model: minimum classification error vs. iterations.

Furthermore, Table 6 provides a summary of experimental performance evaluation factors obtained for the DTDS-BAG-DT model, displaying the values for classification accuracy (ACC), the positive predictive value (PPV), the true positive rate (TPS), the false discovery rate (FDR), the false-negative rate (FNR), the area under the curve (AUC), the classification speed (CS), and the number of misclassified samples (NMS). In addition to the high-performance rates obtained for ACC, PPV, TPR, and AUC, and the low error alarm rates obtained for ERP, FDR, FNR, and NMS, the system exhibits high-speed inferencing for DTDS-BAG-DT, with a CSP value of 110,000 samples per second, scoring a low prediction overhead of 9.09 μ second.

Table 6. Results obtained for optimizable ensemble using bagging decision tree (BAG-DT) model.

ACC	PPV	TPR	HMS	ERP	FDR	FNR	AUC	NMS	CSP
99.50%	99.45%	96.93%	98.18%	0.5%	0.55%	3.07%	100%	668	110,000

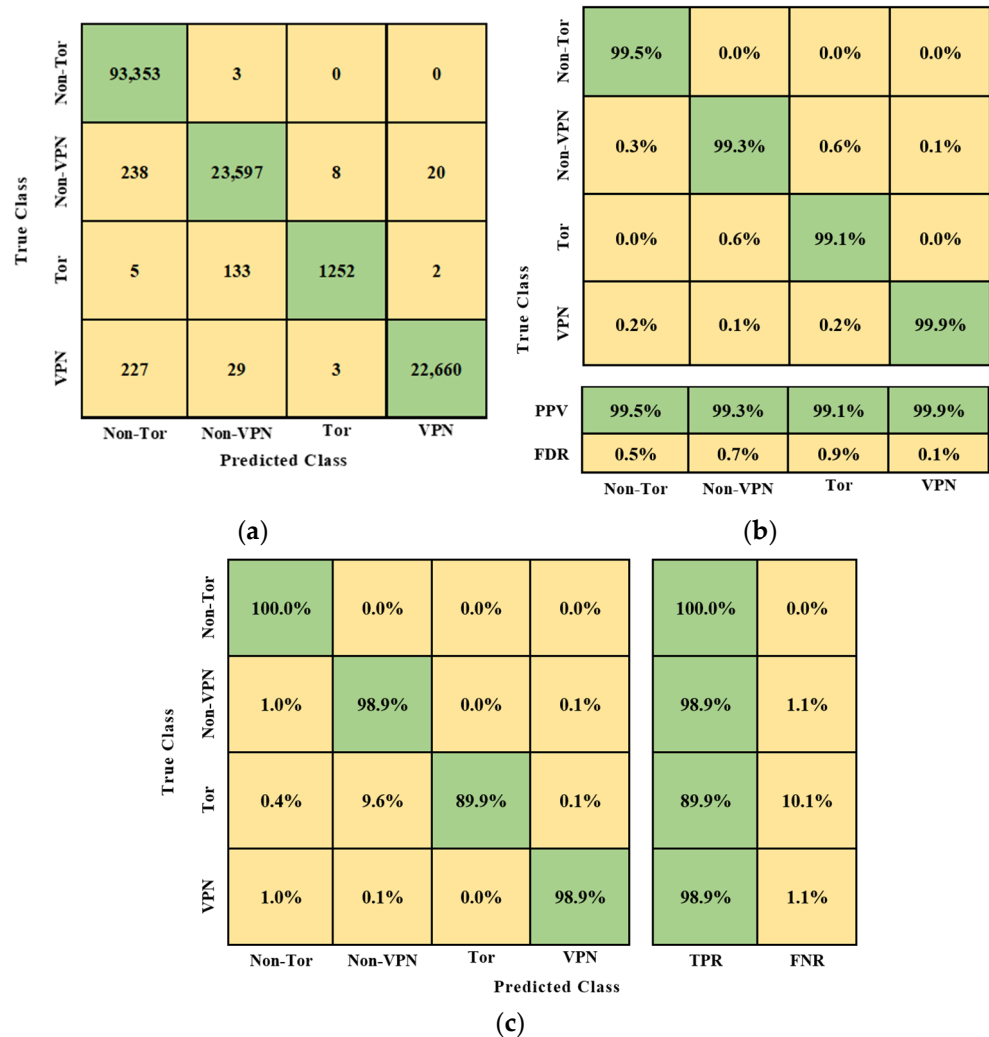


Figure 10. (a) The four-class confusion matrix analysis for the BAG-DT model: true classes vs. predicted classes, (b) The four-class TPR-FNR matrix analysis, for the BAG-DT model: true classes vs. predicted classes, and (c) The four-class PPV-FDR matrix analysis for the BAG-DT model: true classes vs. predicted classes.

Last of all, we benchmark our proposed system with other existing systems. Specifically, in Table 7, we compare our best performance indication results, which correspond to the DTDS-BAG-DT system, with other up-to-date, state-of-the-art systems employing diverse learning approaches (deep and machine learning methods) to detect Darknet traffic activities. The comparison in this table considers the underlying learning method used in each Darknet detection system, the accuracy of detecting/classifying the Darknet activities, and the proportion of improvement factor (%) over the existing system. In this comparison, eleven ML-based DTDS systems are considered in this evaluation, employing different supervised learning systems comprising: recurring neural network (RNN) [1], longitudinal analysis of network traffic (LANT) [2], hierarchical classification method (HCM) [3], AdaBoost decision trees (AB-DT) [4], convolutional neural network (CNN) [5,9], artificial neural network and Apache spark (ANN-AS) [6], hybrid model employing convolutional neural network (CNN) and k-means (KM) [7], sparse structure learning with lasso selection (SSL) [8], random forest classifier (RFC) [10], logistic regression classifier (LRC) [11], and our optimum model involving the bagging decision tree ensembles (BAG-DT) classifier. Based on the information in Table 7, we conclude that our DTDS-BAG-DT model is superior, since it reported the best performance scores among all models in the comparison table. Specifically, our best results are improved by (1.7~27%) over the former state-of-the-art

models. Hence, the developed model can undoubtedly be deployed as an intelligent detection service into the IoT's application layer and routing layer to detect Tor and VPN activities in the IoT network.

Table 7. Comparison with other state-of-the-art models.

Research	Year	Evaluation Model	Accuracy	I.F. %
[49]	2021	Recurring Neural Network (RNN)	94.51%	5.28%↑
[56]	2017	Longitudinal Analysis of Network Traffic (LANT)	94.00%	5.85%↑
[57]	2020	Hierarchical Classification Method (HCM)	96.60%	3.00%↑
[48]	2021	AdaBoost Decision Trees (AB-DT)	97.30%	2.26%↑
[25]	2020	Convolutional Neural Network (CNN)	86.00%	15.70%↑
[33]	2020	Artificial Neural Network and Apache Spark (ANN-AS)	94.66%	5.11%↑
[58]	2021	Convolutional Neural Network (CNN) and K-Means (KM)	97.40%	2.16%↑
[59]	2020	Sparse Structure Learning with LASSO selection (SSL)	97.10%	2.47%↑
[50]	2021	Convolutional Neural Network (CNN)	97.65%	1.89%↑
[60]	2019	Random Forest Classifier (RFC)	78.30%	27.08%↑
[61]	2017	Logistic Regression Classifier (LRC)	96.60%	3.00%↑
Proposed	2022	Bagging Decision Tree Ensembles	99.50%	-

5. Conclusions

An efficient autonomous Darknet traffic detection system (DTDS) has been proposed, modeled, implemented, assessed, and reported in this paper. The proposed system characterizes the performance of six supervised machine-learning techniques, including bagging decision tree ensembles (BAG-DT), AdaBoost decision tree ensembles (ADA-DT), RUS-Boosted decision tree ensembles (RUS-DT), optimizable decision tree (O-DT), optimizable k-nearest neighbor (O-KNN), and optimizable discriminant (O-DSC). The developed DTDS-ML models were evaluated on a modern and inclusive dataset (i.e., CIC-Darknet-2020) involving a large number of captured cyber-attacks and hidden services provided by the Darknet grouped into four classes (VPN, TOR, Non-VPN, Non-TOR). Our work demonstrates that the DTDS-based BAG-DT model is superior among the other evaluated models, scoring 99.5% in classification accuracy with a low inferencing overhead of 9.09 μ second. Furthermore, compared with other state-of-the-art models, our best results have improved the performance of existing DTDS models by a factor of (1.7~27%). Consequently, the proposed model can be efficiently deployed to detect Tor and VPN activities in communication networks.

Author Contributions: Conceptualization, Q.A.A.-H.; methodology, Q.A.A.-H.; software, Q.A.A.-H.; validation, Q.A.A.-H. and M.K.; formal analysis, Q.A.A.-H., M.K. and W.A.E.; investigation, Q.A.A.-H., M.K. and W.A.E.; resources, Q.A.A.-H., M.K. and W.A.E.; data curation, Q.A.A.-H. and W.A.E.; writing—original draft preparation Q.A.A.-H., M.K. and W.A.E.; writing—review and editing, Q.A.A.-H., M.K. and W.A.E.; visualization, Q.A.A.-H., M.K. and W.A.E.; project administration, Q.A.A.-H.; funding acquisition, Q.A.A.-H., M.K. and W.A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Commun. Surv. Tutor.* **2018**, *22*, 1646–1685. [[CrossRef](#)]
- Dastjerdi, A.V.; Buyya, R. Fog Computing: Helping the Internet of Things Realize Its Potential. *Computer* **2016**, *49*, 112–116. [[CrossRef](#)]
- Ray, S.; Jin, Y.; Raychowdhury, A. The Changing Computing Paradigm with Internet of Things: A Tutorial Introduction. *IEEE Des. Test* **2016**, *33*, 76–96. [[CrossRef](#)]
- Lombardi, M.; Pascale, F.; Santaniello, D. Internet of Things: A General Overview between Architectures, Protocols and Applications. *Information* **2021**, *12*, 87. [[CrossRef](#)]
- Čolaković, A.; Hadžialić, M. Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. *Comput. Netw.* **2018**, *144*, 17–39. [[CrossRef](#)]

6. Abu Al-Haija, Q.; Alsulami, A.A. High Performance Classification Model to Identify Ransomware Payments for Heterogeneous Bitcoin Networks. *Electronics* **2021**, *10*, 2113. [[CrossRef](#)]
7. Soro, F.; Drago, I.; Trevisan, M.; Mellia, M.; Ceron, J.; Santanna, J.J. Are Darknets All the Same? On Darknet Visibility for Security Monitoring. In Proceedings of the 2019 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), Paris, France, 1–3 July 2019; pp. 1–6. [[CrossRef](#)]
8. Bertino, E.; Islam, N. Botnets and Internet of Things Security. *Computer* **2017**, *50*, 76–79. [[CrossRef](#)]
9. Koliadis, C.; Kambourakis, G.; Stavrou, A.; Voas, J. DDoS in the IoT: Mirai and Other Botnets. *Computer* **2017**, *50*, 80–84. [[CrossRef](#)]
10. Bergman, M. The Deep Web: Surfacing Hidden Value. Taking License, volume 7, issue 1. August 2001. Available online: <https://quod.lib.umich.edu/j/jep/3336451.0007.104?view=text;rgn=main> (accessed on 13 November 2021).
11. Tor Project. Available online: www.torproject.org (accessed on 21 November 2021).
12. Caspi, G. Introducing Deep Learning: Boosting Cybersecurity with an Artificial Brain. Informa Tech, Dark Reading, Analytics. 2016. Available online: <http://www.darkreading.com/analytics> (accessed on 21 November 2021).
13. Bendiab, G.; Shiaeles, S.; Alruban, A.; Kolokotronis, N. IoT Malware Network Tra_c Classification using Visual Representation and Deep Learning. In Proceedings of the 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, 29 June–3 July 2020; pp. 444–449.
14. Abu Al-Haija, Q.; Zein-Sabatto, S. An Efficient Deep-Learning-Based Detection and Classification System for Cyber-Attacks in IoT Communication Networks. *Electronics* **2020**, *9*, 2152. [[CrossRef](#)]
15. Sapre, S.; Ahmadi, P.; Islam, K. A Robust Comparison of the KDDCup99 and NSL-KDD IoT Network Intrusion Detection Datasets through Various Machine Learning Algorithms. *arXiv* **2019**, arXiv:1912.13204v1.
16. Imamverdiyev, Y.; Sukhostat, L. Anomaly detection in network traffic using extreme learning machine. In Proceedings of the 2016 IEEE 10th International Conference on Application of Information and Communication Technologies (AICT), Baku, Azerbaijan, 12–14 October 2016; pp. 1–4.
17. Albulayhi, K.; Smadi, A.A.; Sheldon, F.T.; Abercrombie, R.K. IoT Intrusion Detection Taxonomy, Reference Architecture, and Analyses. *Sensors* **2021**, *21*, 6432. [[CrossRef](#)]
18. Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 1–21. [[CrossRef](#)]
19. Benkhelifa, E.; Welsh, T.; Hamouda, W. A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3496–3509. [[CrossRef](#)]
20. Abu Al-Haija, Q.; Al Badawi, A.; Bojja, G.R. Boost-Defence for resilient IoT networks: A head-to-toe approach. *Expert Syst.* **2022**, e12934. [[CrossRef](#)]
21. Abu Al-Haija, Q. Top-Down Machine Learning-Based Architecture for Cyberattacks Identification and Classification in IoT Communication Networks. *Front. Big Data* **2022**, *4*, 782902. [[CrossRef](#)] [[PubMed](#)]
22. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access* **2019**, *7*, 82721–82743. [[CrossRef](#)]
23. Abu Al-Haija, Q.; Al-Badawi, A. Attack-Aware IoT Network Traffic Routing Leveraging Ensemble Learning. *Sensors* **2021**, *22*, 241. [[CrossRef](#)]
24. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets, and challenges. *Cybersecurity* **2021**, *4*, 1–27. [[CrossRef](#)]
25. Lashkari, A.H.; Kaur, G.; Rahali, A. DIDarknet: A Contemporary Approach to Detect and Characterize the Darknet Traffic using Deep Image Learning. In Proceedings of the 10th International Conference on Communication and Network Security, Tokyo, Japan, 27–29 November 2020.
26. Fachkha, C.; Debbabi, M. Darknet as a Source of Cyber Intelligence: Survey, Taxonomy, and Characterization. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 1197–1227. [[CrossRef](#)]
27. Early, J.; Brodley, C.; Rosenberg, C. Behavioral authentication of server flows. In Proceedings of the 19th Annual Computer Security Applications Conference, Las Vegas, NV, USA, 8–12 December 2003.
28. Turkett, W.H., Jr.; Karode, A.V.; Fulp, E.W. In-the-Dark Network Traffic Classification Using Support Vector Machines. *AAAI* **2008**, *3*, 1745–1750.
29. Moore, A.W.; Zuev, D. Internet traffic classification using bayesian analysis techniques. In Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Banff, AB, Canada, 6–10 June 2005; pp. 50–60.
30. Wright, C.V.; Monroe, F.; Masson, G.M. On inferring application protocol behaviors in encrypted network traffic. *J. Mach. Learn. Res.* **2006**, *7*, 2745–2769.
31. Erman, J.; Arlitt, M.; Mahanti, A. Traffic classification using clustering algorithms. In Proceedings of the 2006 SIGCOMM Workshop on Mining Network Data, Pisa, Italy, 11–12 September 2006; pp. 281–286.
32. Easttom, W. Virtual Private Networks, Authentication, and Wireless Security. In *Modern Cryptography*; Springer: Cham, Switzerland, 2020; pp. 299–317.
33. Aswad, S.A.; Sonuc, E. Classification of VPN Network Traffic Flow Using Time Related Features on Apache Spark. In Proceedings of the 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, 20–24 October 2020; pp. 1–8.

34. Gupta, A.; Thakur, H.K.; Shrivastava, R.; Kumar, P.; Nag, S. A Big Data Analysis Framework Using Apache Spark and Deep Learning. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 9–16.
35. Bagui, S.; Fang, X.; Kalaimannan, E.; Bagui, S.C.; Sheehan, J. Comparison of machine-learning algorithms for classification of VPN network traffic flow using time-related features. *J. Cyber Secur. Technol.* **2017**, *1*, 108–126. [[CrossRef](#)]
36. Draper-Gil, G.; Lashkari, A.H.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of encrypted and vpn traffic using time-related. In Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP), Rome, Italy, 19–21 February 2016; pp. 407–414.
37. Miller, S.; Curran, K.; Lunney, T. Multilayer Perceptron Neural Network for Detection of Encrypted VPN Network Traffic. In Proceedings of the 2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA), Glasgow, UK, 11–12 June 2018.
38. Varghese, J.E.; Muniyal, B. A Pilot Study in Software-Defined Networking Using Wireshark for Analyzing Network Parameters to Detect DDoS Attacks. In *Information and Communication Technology for Competitive Strategies (ICTCS 2020)*; Springer: Singapore, 2021; pp. 475–487.
39. Basyoni, L.; Fetais, N.; Erbad, A.; Mohamed, A.; Guizani, M. Traffic Analysis Attacks on Tor: A Survey. In Proceedings of the 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT), Doha, Qatar, 2–5 February 2020; pp. 183–188.
40. AlSabah, M.; Bauer, K.; Goldberg, I. Enhancing Tor’s performance using real-time traffic classification. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, Raleigh North, CA, USA, 16–18 October 2012; pp. 73–84.
41. Wang, L.; Mei, H.; Sheng, V.S. Multilevel Identification and Classification Analysis of Tor on Mobile and PC Platforms. *IEEE Trans. Ind. Inform.* **2021**, *17*, 1079–1088. [[CrossRef](#)]
42. Zavrak, S.; Iskefiyeli, M. Anomaly-Based Intrusion Detection from Network Flow Features Using Variational Autoencoder. *IEEE Access* **2020**, *8*, 108346–108358. [[CrossRef](#)]
43. Lingyu, J.; Yang, L.; Bailing, W.; Hongri, L.; Guodong, X. A hierarchical classification approach for tor anonymous traffic. In Proceedings of the 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), Guangzhou, China, 6–8 May 2017; pp. 239–243.
44. Zhao, S.; Zhang, Y.; Chang, P. Network Traffic Classification Using Tri-training Based on Statistical Flow Characteristics. In Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICSS, Sydney, NSW, Australia, 1–4 August 2017; pp. 323–330.
45. Saleh, S.; Qadir, J.; Ilyas, M.U. Shedding Light on the Dark Corners of the Internet: A Survey of Tor Research. *J. Netw. Comput. Appl.* **2018**, *114*, 1–28. [[CrossRef](#)]
46. Sarwar, M.B.; Hanif, M.K.; Talib, R.; Younas, M. DarkDetect: Darknet Traffic Detection and Categorization Using Modified Convolution-Long Short-Term Memory. *IEEE Access* **2021**, *9*, 113705–113713. [[CrossRef](#)]
47. Iliadis, L.A.; Kaifas, T. Darknet Traffic Classification using Machine Learning Techniques. In Proceedings of the 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCASST), Thessaloniki, Greece, 5–7 July 2021; pp. 1–4.
48. Ul Alam, M.Z.; Azizul Hakim, A.; Toufikuzzaman, M. Application and Interpretation of Ensemble Methods for Darknet Traffic Classification. Preprint. In Proceedings of the 42nd IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 24–27 May 2021; IEEE: Piscataway, NJ, USA, 2021.
49. Demertzis, K.; Tsiknas, K.; Takezis, D.; Skianis, C.; Iliadis, L. Darknet Traffic Big-Data Analysis and Network Management for Real-Time Automating of the Malicious Intent Detection Process by a Weight Agnostic Neural Networks Framework. *Electronics* **2021**, *10*, 781. [[CrossRef](#)]
50. Li, Y.; Lu, Y. ETCC: Encrypted Two-Label Classification Using CNN. *Secur. Commun. Netw.* **2021**, *2021*, 6633250. [[CrossRef](#)]
51. Albulayhi, K.; Sheldon, F.T. An Adaptive Deep-Ensemble Anomaly-Based Intrusion Detection System for the Internet of Things. In Proceedings of the 2021 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 10–13 May 2021.
52. Lashkari, A.H.; Gil, G.D.; Mamun, M.S.I.; Ghorbani, A.A. Characterization of Tor Traffic using Time based Features. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy; SCITEPRESS, Porto, Portugal, 19–21 February 2017.
53. Abu Al-Haija, Q.; Al Tarayrah, M.I.; Enshasy, H.M. Time-Series Model for Forecasting Short-term Future Additions of Renewable Energy to Worldwide Capacity. In Proceedings of the 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), Sakheer, Bahrain, 26–27 October 2020.
54. Abu Al-Haija, Q.; Al Nasr, K. Supervised Regression Study for Electron Microscopy Data. In Proceedings of the 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, 18–21 November 2019; pp. 2661–2668. [[CrossRef](#)]
55. Abu Al-Haija, Q.; Smadi, A.A.; Allehyani, M.F. Meticulously Intelligent Identification System for Smart Grid Network Stability to Optimize Risk Management. *Energies* **2021**, *14*, 6935. [[CrossRef](#)]
56. Liu, J.; Fukuda, K. An Evaluation of Darknet Traffic Taxonomy. *J. Inf. Process.* **2018**, *26*, 148–157. [[CrossRef](#)]
57. Hu, Y.; Zou, F.; Li, L.; Yi, P. Traffic Classification of User Behaviors in Tor, I2P, ZeroNet, Freenet. In Proceedings of the 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, 29 December 2020–1 January 2021; pp. 418–424. [[CrossRef](#)]

58. Li, Y.; Lu, Y.; Li, S. EZAC: Encrypted Zero-day Applications Classification using CNN and K-Means. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; pp. 378–383. [[CrossRef](#)]
59. Han, C.; Shimamura, J.; Takahashi, T.; Inoue, D.; Takeuchi, J.I.; Nakao, K. Real-Time Detection of Global Cyberthreat Based on Darknet by Estimating Anomalous Synchronization Using Graphical Lasso. *IEICE Trans. Inf. Syst.* **2020**, *E103-D*, 2113–2124. [[CrossRef](#)]
60. Zenebe, A.; Shumba, M.; Carillo, A.; Cuenca, S. Cyber Threat Discovery from Dark Web. *EPiC Ser. Comput.* **2019**, *64*, 174–183. [[CrossRef](#)]
61. Al Nabki, M.W.; Fidalgo, E.; Alegre, E.; De Paz, I. Classifying Illegal Activities on Tor Network Based on Web Textual Contents. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, 3–7 April 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 35–43.